# Modelling Biological Sequences by Grammatical Inference

François Coste

Symbiose Project
IRISA/INRIA Rennes-Bretagne Atlantique, France

## Foreword

This document is a complement of the tutorial on "Modelling Biological Sequences by Grammatical Inference" organized for the tenth anniversary edition of the International Colloquium on Grammatical Inference (ICGI 2010) held in Valencia, Spain.

The tutorial surveys the approaches related to grammatical inference which have been developed in Bioinformatics to model family of sequences[1], from well established weighting schemes for profile hidden Markov models and Stochastic Context-Free grammars (section 2) to approaches learning (also) the structure or topology of the grammars (section 3). The intent of this document is to give bibliographic references of original studies presented during the tutorial and to provide helpful pointers to start a deeper investigation into this field. This is also the opportunity to gather a list of published work related to this topic. The list is obviously not exhaustive but can be the starting point of a more exhaustive one: let me know the references that you know on the subject, which are neither cited in this document nor in the cited papers, and I will be pleased to complete this bibliography and to maintain an updated version which will be available from my homepage (`http://www.irisa.fr/symbiose/francois_coste`).

## 1 Linguistic modelling of biological sequences

Recent sequencing projects and technological progress in the field are giving access to an ever increasing amount of DNA, RNA and protein macromolecular sequences. The challenge in the post-genomic era is now to decipher what has been popularly named "the language of life" [2]. The linguistic metaphor has been used indeed for a long time in molecular biology, and applying computational linguistics tools to represent, understand and handle biological sequences is a natural continuation of this metaphor. Using formal grammars, such as the ones

[1] Let us remark that the focus of the tutorial is on the inference of *generative* (grammatical) models. This excludes other popular machine learning methods which have been successful in Bioinformatics, especially the *discriminative* ones such as the neural networks and support vector machines which have been shown to be very efficient even for sequence classification tasks (see for instance [1]).

introduced by Noam Chomsky [3] to describe natural languages and to study syntax acquisition by children, has been advocated in particular by Searls [4, 5] whose articles provide a good introduction to the different levels of expressiveness required to model macromolecular sequences and to related grammatical formalisms. The problem is then the classical one of finding a good trade-off between the expressiveness of the grammar and the efficiency of the dedicated parser used on genomic sequences to retrieve the sequences belonging to the family modelled by the grammar or to analyse the structure of these sequences. The main difficulty in this approach is that, in contrast with all the studies available on natural languages, little is known about the syntax of DNA, as shown for instance, by the lack of reliable definitions of "words", "sentences" or "punctuation marks". Building the grammars is thus not an easy task. In some cases expert knowledge allows to design the grammars, usually by a succession of trial-and-error refinements until a satisfactory model is found. More often, available expert knowledge is not sufficient. In the tutorial, we focus on machine learning approaches to help building the grammars. Instead of asking the expert to design the grammar, the idea is rather to ask the expert to provide a set of characteristic sequences of the family to model and to let a *grammatical inference* program discover by induction the best grammar modelling these sequences.

When considering biological sequences, the induction relies generally on the search of *conservation*. In biology, genetic variation (by mutations, recombination of chromosomes, crossing-over and other sources of sequence variation) is a fundamental source of diversity which is opposed to the natural selection. A conserved feature among a set of sequences is an evidence of selection through evolution and is thus likely to be important for the family. This principle underlies many of the approaches presented in the tutorial.

The intrinsic variability of the sequences has also to be handled by the grammatical models, a usual solution being to use *stochastic* models. For such grammars, the inference problem has then two aspects: determining the structure (topology) of the grammar and estimating the probabilistic parameters. We begin in section 2 with the parameter estimation problem, the structure being given, before considering the problem of learning (also) the structure or topology of the grammars in section 3.

## 2 Parameter estimation

The parameter estimation approach has been particularly successful in Bioinformatics. The principle is to choose a generic simple grammar topology which is sufficient to capture the principal characteristics of the sequences to model – for instance, profile hidden Markov models (pHMM) for proteins or their context-free or tree adjoining grammars equivalent for RNA sequences without or with pseudo-knots – and to fit the parameter probabilities (or weights) to maximize the likelihood of the available sample of sequences from the family. Two starting points are recommended to discover this approach: "Biological Sequence Analysis" by Durbin, Eddy, Krogh and Mitchinson [6] is a well known reference al-

lowing to investigate profile hidden Markov models, and the more recent survey by Sakakibara [7] with a greater grammatical flavor allows in particular to learn more about the stochastic context-free grammars used to model RNA sequences.

These approaches have been very successful in Bioinformatics, and several databases containing this kind of signatures are publicly available on the web and commonly used by biologists. One can cite PFAM [8] modelling protein domain families by pHMM and RFAM [9] modeling common non-coding RNA families by stochastic context-free grammars named covariance models. Two main packages are available and commonly used for building and using pHMM : HMMR [10] and SAM [11]. One has to note however, that the number of sequences available for training this kind of models is usually small compared to the number of parameters, even if special attention has been given to diminish their number by the design and the choice of simple topologies. In practice, as illustrated by the ISMB'99 SAM's tutorial (`http://compbio.soe.ucsc.edu/ismb99.tutorial.html`), modelling proteins relies rather on multiple sequence alignments tools [12] and essentially on powerful pseudo-counts weighting schemes based on prior knowledge about mutation preferences of amino-acids (see for instance [13]) than on the training sample. A good example of this is that in SAM's tutorial, the classical work-flow starts with a unique sequence in the training set. A similar tendency towards using more and more priors can also be observed for modelling RNA.

In these approaches, the general topology is usually fixed. One can still play with the number of matching states in pHMM or their equivalent in stochastic context free formalisms to test topologies of different lengths. These numbers can be tuned manually by testing the performance of the different models after training. It can also constitute parameters to optimize in the Bayesian framework. Recent promising work on the prediction of RNA secondary structure [14] allows to get rid of these parameters by extending classical stochastic context free grammars used traditionally to model RNA to HDP-SCFGs which employ an infinite set of nonterminals thanks to a hierarchical Dirichlet process.

In the following section, we focus on approaches learning the topology of the grammar.

## 3 Grammar Inference

Learning the characteristic syntax of a family of sequences is a difficult task. Pattern (or Motif) Discovery is an important and active field of Bioinformatics. It aims at finding a common pattern present in a family of sequences. The patterns used range from simple words to sub-regular expressions, allowing to express the choice of several letters at one positions and rigid or variable length gaps between positions. Good reviews of the field can be found in [15] and [16]. Among the state-of-the-art algorithms that learn expressive patterns, one can mention Pratt [17] (chosen to be the default pattern discovery tool proposed on the Prosite database of protein domains, families and functional sites [18]), EMotif Maker [19], Teiresias [20] and Splash [21]. The expressiveness of the patterns that are

learnt by these methods is still below the simplest level of Chomsky hierarchy; in particular such patterns do not allow to express correlation between the choice of letters between positions.

A first step toward learning grammars is the early work of Yokomori [22] on learning locally $k$-testable languages for the identification of protein $\alpha$-chain regions. These languages can be represented by a subclass of automata, which are linked to *n-grams* and, more biologically, to persistent splicing systems. Locally $k$-testable languages have the property that it is sufficient to parse substrings of length $k$ to decide whether a sequence is accepted or not. This method is thus restricted to local characterizations of length $k$, which has furthermore to be fixed usually to a small value to avoid over-specialization by the inference algorithm. To allow the application to proteins whose alphabet contains 20 letters, amino-acid recoding was used in a two letters alphabet (hydrophobic or polar) or in 7 letters alphabet (Dayhoff encoding). This work is the root of recent studies, applying successfully similar approaches to the prediction of coiled-coil proteins [23] and to the prediction of transmembrane domains in proteins [24].

At the first level of Chomsky's hierarchy (regular languages), Protomata-Learner [25, 26] is a successful application of the classical state merging framework developed in grammatical inference, to learning automata on functional or structural families of proteins. It introduces a new kind of alignment, named partial local multiple alignment (PLMA), which is better suited than classical multiple sequence alignment to the expressiveness of automata: by merging blocks of conservation identified in the PLMA, an automaton with alternative paths and thus correlations between positions can be learnt. Protomata-Learner offers the opportunity to learn more expressive topologies than the pHMM ones, while still benefiting from the weighting schemes developed for pHMM: this allows to model, eventually heterogeneous, families of proteins with a finer level of precision.

Regular grammar topologies or even less expressive formalisms can be sufficient to characterize protein families in many cases, but they cannot model (potentially nested or crossing) long-term dependencies such as contacts of amino-acids that are far in the sequence but close in the 3D folding of the protein. In [27], the authors propose a framework, based on the combination of stochastic context free grammars related to different physico-chemical properties of the amino acids and on genetic algorithms, that is shown to produce relevant protein binding site descriptors.

Inference of context free grammars has also been applied to DNA sequences. The main line of research in this area, initiated by Sequitur [28], is to learn the hierarchical (context free) structure of a (long) DNA sequence. Following Occam's razor principle, this task can be formalized as the problem of finding the smallest context free grammar generating the sequence. This problem lies in the intersection of several communities as it is linked to Kolmogorov complexity estimation, data compression and grammatical inference. In particular, it can be seen as the inference of a grammar, the language being known and restricted to the given sequence, or as a first step of the inference, consisting in

the identification of the unlabeled derivation tree of the sequence by a context free grammar to be learnt afterwards. Sequitur is a fast on-line algorithm that constructs a grammar by reading the sequence from left to right, by replacing any digram repetition with a non terminal producing it and by ensuring that each rule is used more than once. This simple, generic scheme has been applied to DNA sequences and was able to beat gzip and PPMC in compression rate [29], while quality is still hard to compare in this domain. DNASequitur [30] improves slightly the performance of Sequitur by adding the reverse complement as a source of repetition. The left to right on-line strategy of Sequitur biases strongly the shape of the structures found. Off-line algorithms have been designed and tested on DNA sequences. GTAC [31] proposes to replace the longest repeated word first and has been used to estimate the entropy of DNA sequences. Smaller grammars can be obtained by spending more time to choose the repeats that are rewritten by non terminals: the strategy of selecting the repeats that greedily compress the grammar best is particularly efficient for the compression of biological sequences [32–34]. Considering the choice of the repeats but also the choice of their occurrences [35] opens new perspectives: experiments on whole genomes shows that grammars up to 9% smaller than the best competitors can be found by this kind of algorithms [36]. Specialisation of the algorithm to the characteristic features of DNA (reverse complement, double strand and sequence mutations) is a work in progress which should help finding even smaller grammars. Asserting and validating from a biological perspective the quality of such grammars will pose further challenges, since a reference grammar is missing...

## References


1. Baldi, P., Brunak, S.: Bioinformatics: The Machine Learning Approach. 2nd edn. Cambridge: MIT Press (2001)
2. Beadle, G.W., Beadle, M.: The language of life: an introduction to the science of genetics. American Institute of Biological Sciences (1966)
3. Chomsky, N.: Syntactic Structures. Mouton & Co. (1957)
4. Searls, D.B.: The language of genes. Nature **420** (2002) 211–217
5. Chiang, D., Joshi, A.K., Searls, D.B.: Grammatical representations of macromolecular structure. Journal of Computational Biology **13** (2006) 1077–1100
6. Durbin, R., Eddy, S., Krogh, A., Mitchison, G.: Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press (1999)
7. Sakakibara, Y.: Grammatical inference in bioinformatics. IEEE Trans. Pattern Anal. Mach. Intell. **27** (2005) 1051–1062
8. Sammut, S.J., Finn, R.D., Bateman, A.: Pfam 10 years on: 10 000 families and still growing. Briefings in Bioinformatics **9** (2008) 210–219
9. Gardner, P.P., Daub, J., Tate, J.G., Nawrocki, E.P., Kolbe, D.L., Lindgreen, S., Wilkinson, A.C., Finn, R.D., Griffiths-Jones, S., Eddy, S.R., Bateman, A.: Rfam: updates to the RNA families database. Nucl. Acids Res. **37** (2009) D136–140
10. S., E.: Hmmer user's guide: biological sequence analysis using prole hidden markov models. http://hmmer.wustl.edu/ (1998)
11. Karplus, K.: Hidden markov models for detecting remote protein homologies. Bioinformatics **14** (1998) 846–865

12. Edgar, R.C., Batzoglou, S.: Multiple sequence alignment. Current Opinion in Structural Biology **16** (2006) 368 – 373 Nucleic acids/Sequences and topology - Anna Marie Pyle and Jonathan Widom/Nick V Grishin and Sarah A Teichmann.
13. Brown, M., Hughey, R., Krogh, A., Mian, I.S., Sjölander, K., Haussler, D.: Using dirichlet mixture priors to derive hidden markov models for protein families. In Hunter, L., Searls, D.B., Shavlik, J.W., eds.: Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology, Bethesda, MD, USA, July 1993, AAAI (1993) 47–55
14. Sato, K., Hamada, M., Mituyama, T., Asai, K., Sakakibara, Y.: A non-parametric bayesian approach for predicting RNA secondary structures. In Salzberg, S., Warnow, T., eds.: Algorithms in Bioinformatics, 9th International Workshop, WABI 2009, Philadelphia, PA, USA, September 12-13, 2009. Proceedings. Volume 5724 of Lecture Notes in Computer Science., Springer (2009) 286–297
15. Brejova, B., Vinar, T., Li, M.: Pattern Discovery: Methods and Software. In Krawetz, S.A., Womble, D.D., eds.: Introduction to Bioinformatics. Humana Press (2003) 491–522
16. Brazma, A., Jonassen, I., Vilo, J., Ukkonen, E.: Pattern discovery in biosequences. In Honavar, V., Slutzki, G., eds.: ICGI. Volume 1433 of Lecture Notes in Computer Science., Springer (1998) 257–270
17. Jonassen, I.Collins, J., Higgins, D.: Finding flexible patterns in unaligned protein sequences. Protein Science **4** (1995) 1587–1595
18. Sigrist, C.J.A., Cerutti, L., Hulo, N., Gattiker, A., Falquet, L., Pagni, M., Bairoch, A., Bucher, P.: PROSITE: A documented database using patterns and profiles as motif descriptors. Brief Bioinform **3** (2002) 265–274
19. Nevill-Manning, C., Wu, T., Brutlag, Douglas, L.: Highly specific protein sequence motifs for genome analysis. PNAS **95** (1998) 5865–5871
20. Rigoutsos, I., Floratos, A.: Combinatorial pattern discovery in biological sequences: the TEIRESIAS algorithm. Bioinformatics **14** (1998) 55–67
21. Califano, A.: Splash: structural pattern localization analysis by sequential histograms. Bioinformatics **16** (2000) 341–357
22. Yokomori, T., Ishida, N., Kobayashi, S.: Learning local languages and its application to protein $\alpha$-chain identification. In: HICSS (5). (1994) 113–122
23. Peris, P., López, D., Campos, M., Sempere, J.M.: Protein motif prediction by grammatical inference. In Sakakibara, Y., Kobayashi, S., Sato, K., Nishino, T., Tomita, E., eds.: ICGI. Volume 4201 of Lecture Notes in Computer Science., Springer (2006) 175–187
24. Peris, P., López, D., Campos, M.: Igtm: An algorithm to predict transmembrane domains and topology in proteins. BMC Bioinformatics **9** (2008)
25. Coste, F., Kerbellec, G.: A similar fragments merging approach to learn automata on proteins. In Gama, J., Camacho, R., Brazdil, P., Jorge, A., Torgo, L., eds.: ECML. Volume 3720 of Lecture Notes in Computer Science., Springer (2005) 522–529
26. Burgos, A., Coste, F., Kerbellec, G.: Learning automata on protein sequences by partial multiple sequence alignement. (in preparation)
27. Dyrka, W., Nebel, J.C.: A stochastic context free grammar based framework for analysis of protein sequences. BMC Bioinformatics **10** (2009) 323
28. Nevill-Manning, C.G., Witten, I.H.: Compression and explanation using hierarchical grammars. The Computer Journal **40** (1997) 103–116
29. Nevill-Manning, C.G., Witten, I.H.: Compression and explanation using hierarchical grammars. Comput. J. **40** (1997) 103–116

30. Cherniavsky, N., Lander, R.: Grammar-based compression of DNA sequences. In: DIMACS Working Group on The Burrows-Wheeler Transform. (2004) 21
31. Lanctot, J.K., Li, M., Yang, E.H.: Estimating DNA sequence entropy. In: ACM-SIAM Symposium on Discrete Algorithms. (2000) 409–418
32. Apostolico, A., Lonardi, S.: Off-line compression by greedy textual substitution. Proceedings of the IEEE **88** (2000) 1733–1744
33. Apostolico, A., Lonardi, S.: Compression of biological sequences by greedy off-line textual substitution. In: Data Compression Conference. (2000) 143–153
34. Nevill-Manning, C., Witten, I.: On-line and off-line heuristics for inferring hierarchies of repetitions in sequences. In: Data Compression Conference, IEEE (2000) 1745–1755
35. Carrascosa, R., Coste, F., Gallé, M., López, G.G.I.: Choosing word occurrences for the smallest grammar problem. In Dediu, A.H., Fernau, H., Martín-Vide, C., eds.: LATA. Volume 6031 of Lecture Notes in Computer Science., Springer (2010) 154–165
36. Carrascosa, R., Coste, F., Gallé, M., Infante-Lopez, G.: Searching for smallest grammars on dna sequences. (submitted)